



APPLIED
SCIENCES
FACULTY ●

Optimization of the Transformer's attention

Linear Algebra (1.22-23.PKN22/M)

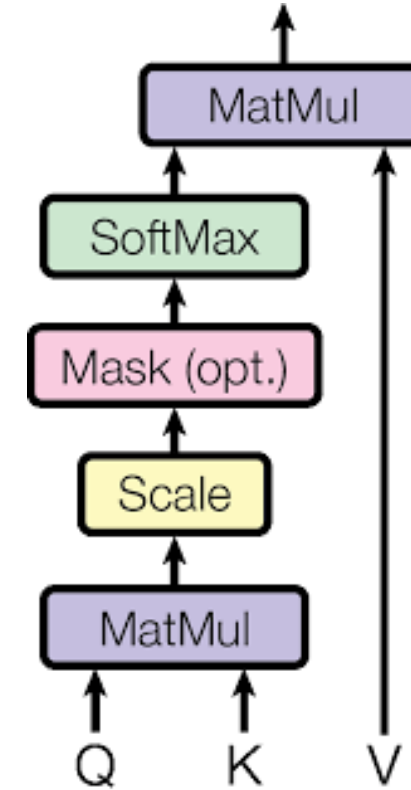
Andrii Ruda

Anton Brazhniy

Oleksandr Korniiienko

Project goals

1. Investigate method of improvement for dot-product (softmax) attention.
2. Implement and evaluate method on CIFAR-10 image classification benchmark.



CIFAR-10 banchmark: Learning multiple layers of features from tiny images.

Technical report, University of Toronto, 2009

Image source: <https://arxiv.org/abs/1706.03762>

Attention Is All You Need

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$X \in \mathbb{R}^{batch \times tokens \times d_k}, Q = XW^Q, K = XW^K, V = XW^V$$

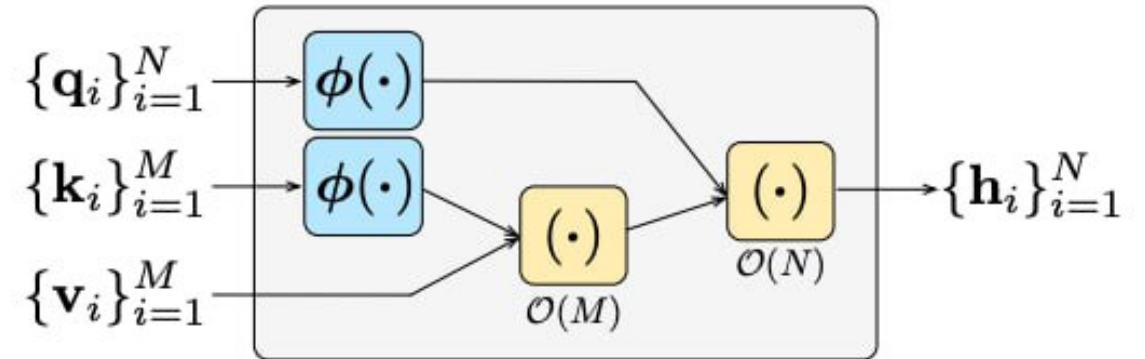
- d_{model} is the size of the embedding vector of each input element from our sequence.
- d_k is the inner dimension of that is specific to each self-attention layer.
- *batch* is the batch size
- *tokens* is the number of elements that our sequence has, e.g. number of pixels.

Source: <https://theaisummer.com/self-attention>

Kernel-based attention optimization methods

Random feature attention

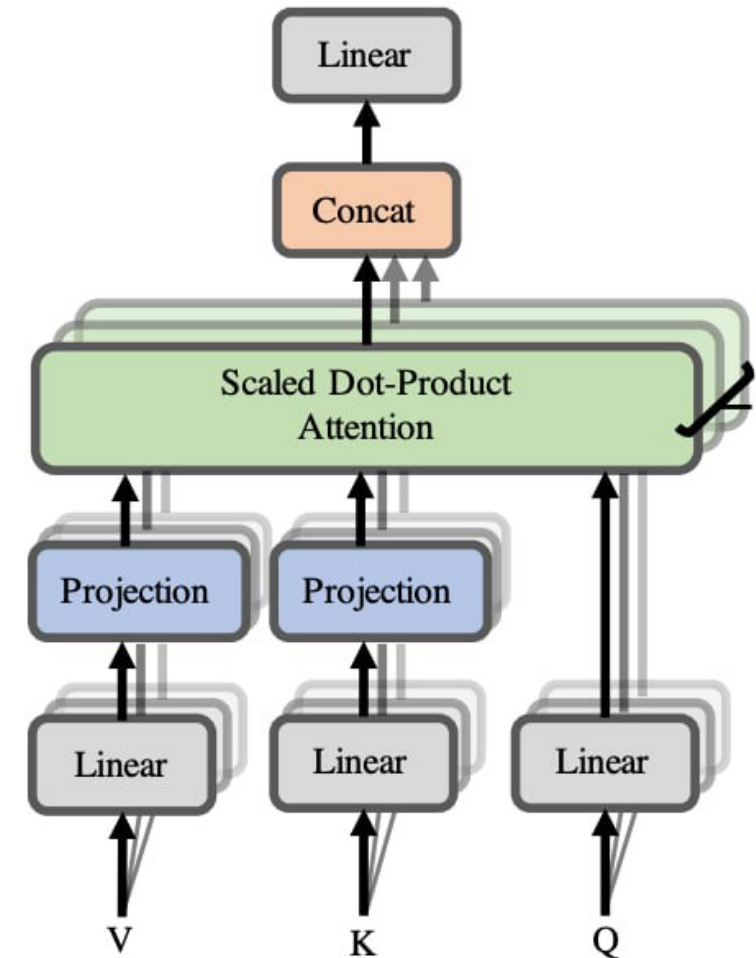
$$P = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \approx \frac{\phi(q)^T \sum_i \phi(k_i) \otimes v_i}{\phi(q) \cdot \sum_j \phi(k)}$$

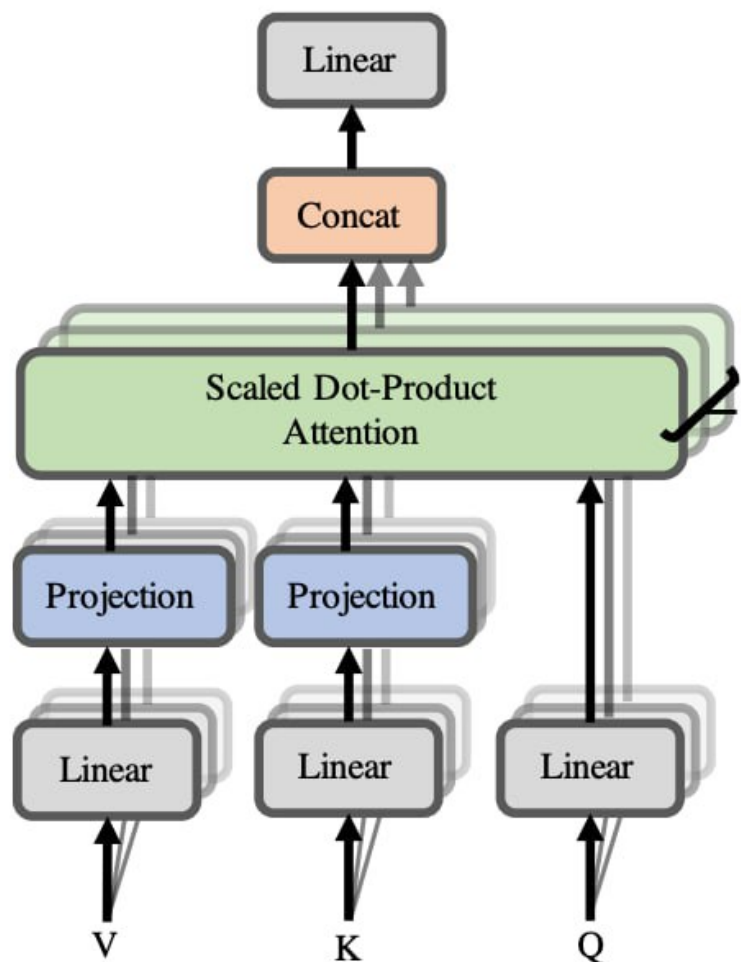


Low-rank attention optimization methods

Linear Attention

$$\begin{aligned}\text{Linear Attention}(Q, K, V) &= \\ &= \left(\frac{Q}{\sqrt{(d_k)}} \right) \left(\frac{K^T}{\sqrt{(d_k)}} V \right) = \\ &= \frac{1}{d_k} Q (K^T V) = \frac{1}{d_k} (Q K^T) V = \\ &\text{Attention}(Q, K, V)\end{aligned}$$





Low-rank attention optimization methods

Linformer Attention

$$\begin{aligned} \text{Linformer}(Q, K, V) &= \\ &= \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \approx \\ &\approx \text{Softmax}\left(\frac{Q(EK)^T}{\sqrt{d_k}}\right)(FV) \end{aligned}$$

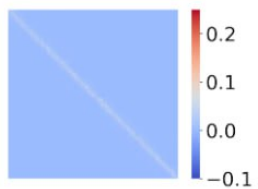
Low-rank attention optimization methods

Nystrom Attention

$$S = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right) = \begin{bmatrix} A_S & B_S \\ F_S & C_S \end{bmatrix}$$

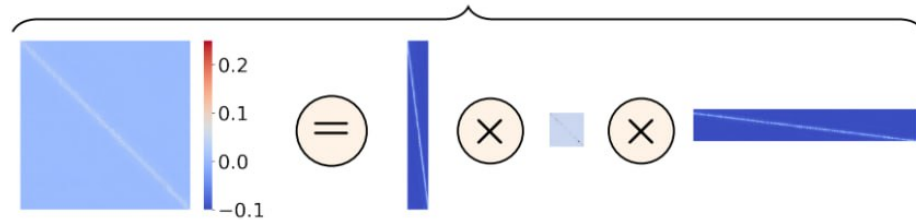
$$\hat{S} = \text{softmax}\left(\frac{Q\tilde{K}^T}{\sqrt{d_q}}\right) \left(\text{softmax}\left(\frac{\tilde{Q}\tilde{K}^T}{\sqrt{d_q}}\right)\right)^\dagger \text{softmax}\left(\frac{\tilde{Q}K^T}{\sqrt{d_q}}\right)$$

softmax

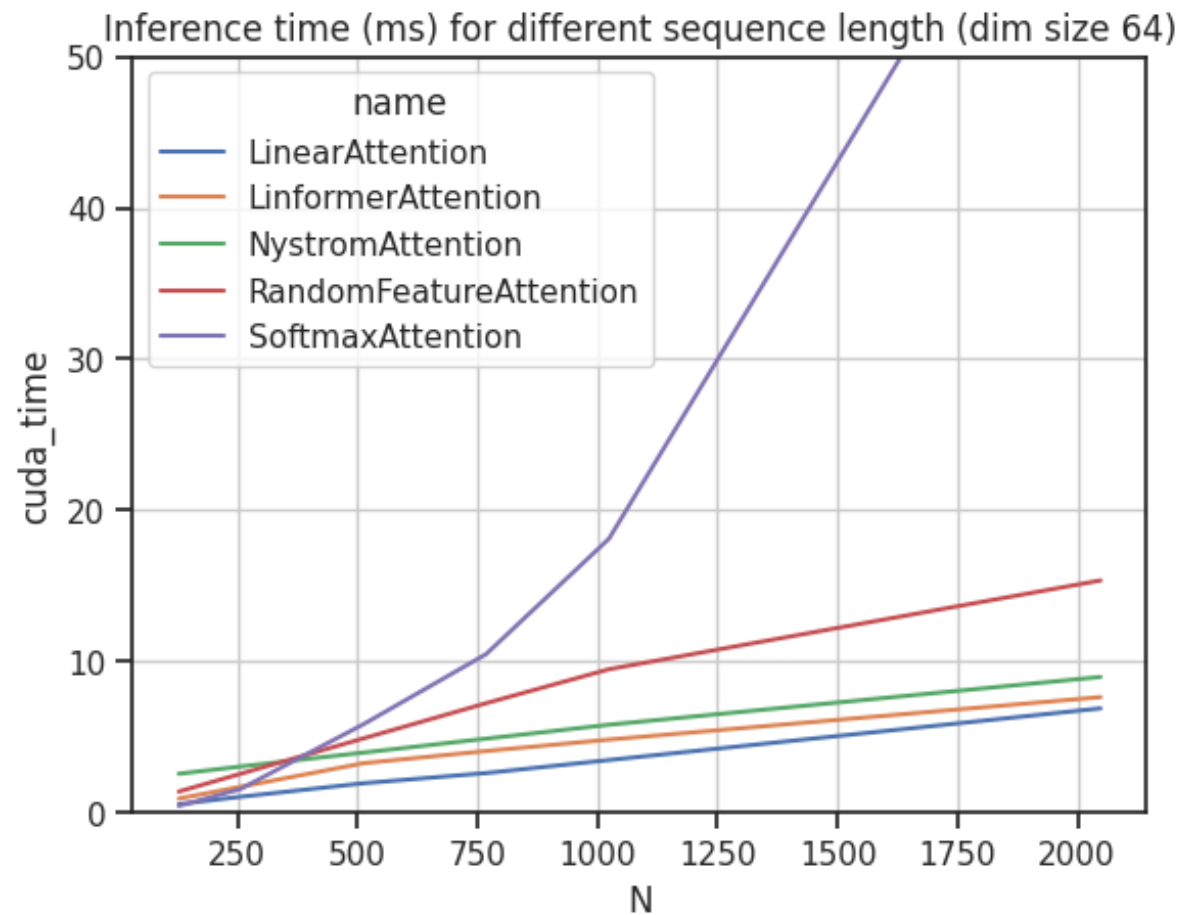


\approx

Nyström approximation



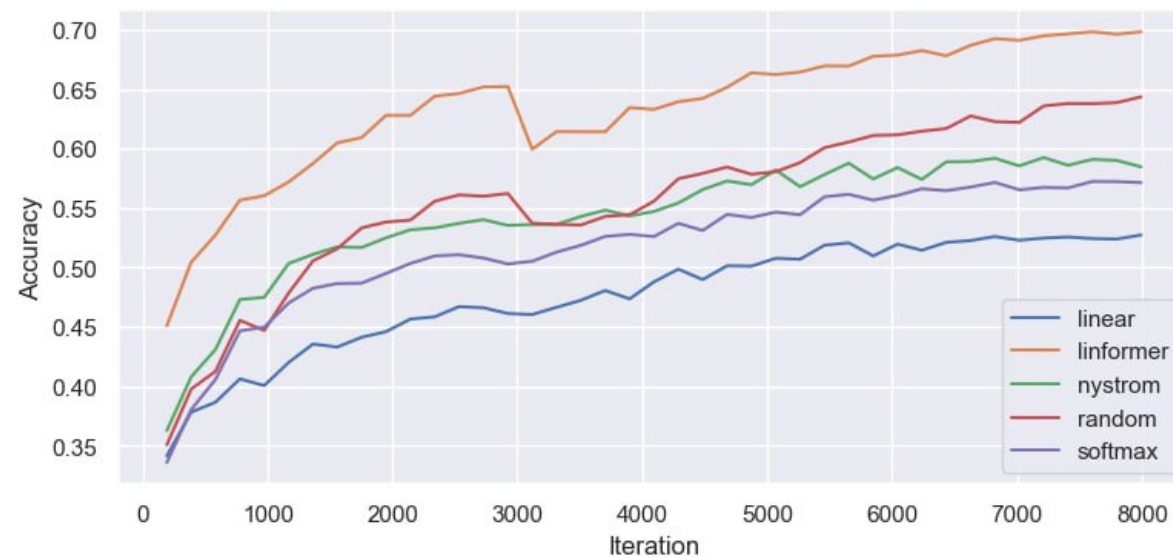
Attention Complexity



Attention Complexity

Method	Computational	Memory
Dot-product (softmax) attention	$O(d^2n + dn^2)$	$O(nd^2)$
Linear attention	$O(d^2n)$	$O(dn + d^2)$
Linformer Attention	$O(n)$	$O(n)$
Random Feature Attention	$O(nd)$	$O(4D + 2Dd)$
Nystrom Attention	$O(n)$	$O(n)$

Attention accuracy



Conclusions

- Investigated dot-product attention mechanism optimization using **linear algebra** matrix transformation techniques.
- Empirically demonstrated that linear attention approach has lower computational complexity than observed methods.
- Linformer attention mechanism showed best classification accuracy among observer models with similar hyper-parameter sets.

Thanks for your attention!

